

Best Practice: Avails, ALIDs and Manifest Delivery Use Cases

DRAFT

CONTENTS

1	Introduction	5
1.1	References.....	5
1.2	Document Conventions.....	5
2	Connecting Objects.....	6
2.1	Mapping Avail to Experience.....	6
2.2	Constructing ALID.....	7
2.3	Mapping Avail file references to Media Manifest asset references	7
2.3.1	Referencing assets by identifier.....	8
2.3.2	Referencing assets by location	8
2.3.3	Images	9
2.4	Determining Which Tracks Are Included in an Entitlement	10
3	Movie	12
3.1	Movie	12
3.2	Regional Edits.....	12
3.2.1	Static View (all edits known)	12
3.2.2	Dynamic View (edits arrive over time).....	13
3.2.3	Regional Manifests	14
3.3	Director's Cut (or other special cut).....	15
3.3.1	Distinct products	15
3.3.2	Directors' Cut as bonus.....	15
3.3.3	Director's Cut as Bundle	16
3.4	Movie with Bonus.....	16
3.5	With and without bonus.....	18
3.6	Movie with Extras, Avails reference Extras	18
4	Episodic	20
4.1	Episode.....	20
4.2	Episode with Bonus	20
4.3	Season.....	20
4.3.1	Without episodes listed (season pass, some or all episodes not aired)	20
4.3.2	With episodes listed (season aired)	21
4.4	Miniseries.....	22
4.5	Alternate Episode Ordering.....	23
4.6	Series, Season and Episode with bonus.....	24
5	Bundles	25
5.1	Movie Bundle	26
5.2	Season Bundle.....	27
5.3	Mixed Bundle	28
5.4	Bundle of Bundles.....	28



This work is licensed under a [Creative Commons Attribution 3.0 Unported License](https://creativecommons.org/licenses/by/3.0/).

NOTE: No effort is being made by the Motion Picture Laboratories to in any way obligate any market participant to adhere to this specification. Whether to adopt this specification in whole or in part is left entirely to the individual discretion of individual market participants, using their own independent business judgment. Moreover, Motion Picture Laboratories disclaims any warranty or representation as to the suitability of this specification for any purpose, and any liability for any damages or other harm you may incur as a result of subscribing to this specification.

DRAFT

REVISION HISTORY

Version	Date	Description
1.0		Initial release

DRAFT

1 INTRODUCTION

This document describes best practices for addressing specific delivery use cases. This is a mostly illustrated guide for structure of Avails and Media Manifest.

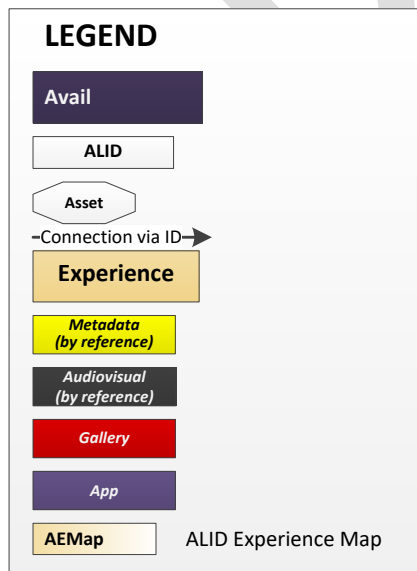
It is applicable to: Avails, Common Metadata, Media Entertainment Core (MEC), Media Manifest, Media Manifest Core (MMC), Cross-Platform Extras (CPE) and other MovieLabs Digital Distribution Framework (MDDF) Specs. This document assumes familiarity with the referenced specifications.

1.1 References

[CM]	Common Metadata, TR-META-CM, www.movielabs.com/md/md
[Manifest]	MovieLabs Common Media Manifest Metadata v1.4, TR-META-MMM, www.movielabs.com/md/manifest
[Avail]	EMA Content Availability Data (Avails), TR-META-AVAIL, www.movielabs.com/md/avails
[MEC]	Media Entertainment Core, TR-META-MEC, www.movielabs.com/md/mec

1.2 Document Conventions

The following shows the drawing conventions used in this document:



2 CONNECTING OBJECTS

This section describes how one gets from Avail to Media Manifest and from Media Manifest to physical (encoded) assets.

The mappings can be simple and direct. However, in practice information is often incomplete or inconsistent. This section attempts to address the real-world scenarios to allow mapping under various non-ideal conditions.

2.1 Mapping Avail to Experience

The most direct means to map Avail to Experience is to use the Avail Experience mapping found in the Media Manifest (MediaManifest/ALIDExperienceMap). This defines the set of Experiences that satisfy a set of Avails.



Selection of the precise Experience then involves looking for the Experience that matches language and region as specified in the Avail. Information on region can be found in Section **Error! Reference source not found.** In XML, the ALID is in Avail/ALID. In Excel, the ‘ALID’ is determined as follows:

Type of Avail	Excel 1.6, when using EIDR IDs	Excel 1.7, when using EIDR IDs	Excel, if using custom ID equivalent
Movie/Episode	ProductID	EditID	AltID
Season	SeasonContentID	SeasonID	SeasonAltID
Series	SeriesContentID	SeriesID	SeriesAltID

If, for some reason, the three elements are not simultaneously available it is still possible to infer which Experience applies to which Avail. However, this is not a general solution and might not work in all circumstances. The Retailer and the Distribution Entity organization must agree upon constraints that allow this work.

Following are some examples of what could work *under controlled circumstance*:

- Assuming only a single asset and one release per region, Experience could be matched based on Avail/@ContentID matching Experience/BasicMetadata/@ContentID, and region/territory matching (see Best Practice). One could, by convention, use Avail/Asset/Metadata/ProductID to match Experience/BasicMetadata/@ContentID.
- Assuming multiple assets and one release per region, Experience can be matched based on ProductID (1.6) or EditID (1.7) as in the previous example. One would have to

identify an Experience that contained all the ProductIDs from all the Assets in the Avail. This would require constraints on the Avail to avoid assets mapping to more than one Experience.

2.2 Constructing ALID

The combination of ALID and Licensor SHALL be globally unique. ALID SHOULD be globally unique. A Licensor SHALL NOT use an ALID for a different combination of assets and terms. An Avail associated with an ALID SHALL have only one disposition.

An ALID SHOULD comply with the ALID format above. This is not a strict requirement, but it will make global uniqueness much easier and avoid us IDs in the wrong context. Note that UUIDs avoid the first issue, but not the second.

An ALID SHOULD be based on an EIDR ID.

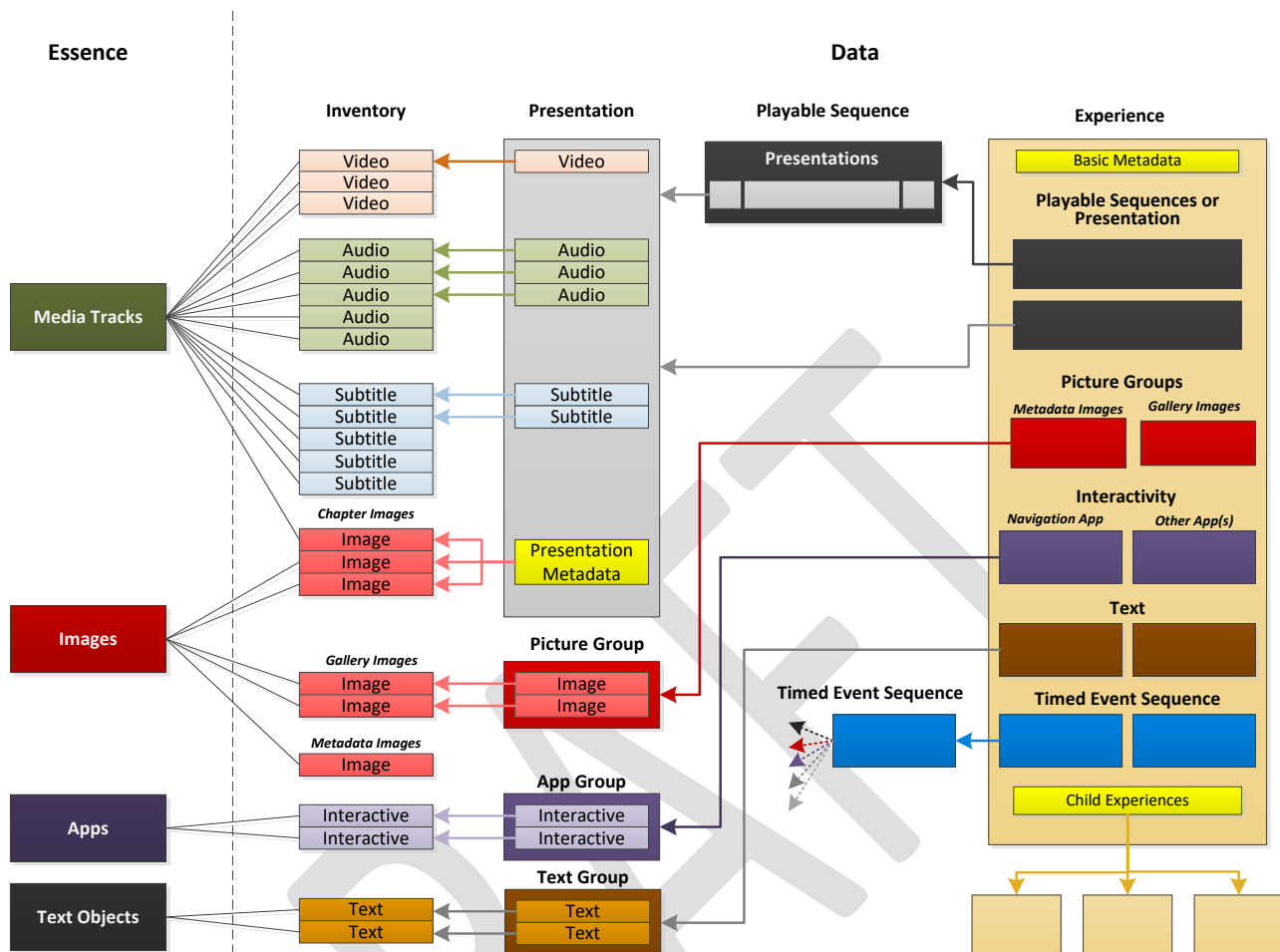
An ALID defines the product. See Section 2.1 for information on deriving ALID from Excel Avails. Accordingly, the same ALID may be used in more than one Avail.

When an ALID refers to a single asset (e.g., a movie, a TV season or a TV episode), it should contain the same EIDR ID as the asset's ContentID.

2.3 Mapping Avail file references to Media Manifest asset references

The Media Manifest provides a very direct means to map Experience to Assets.

In the following diagram, the vertical dotted line shows where this mapping occurs. This process is described in detail in [Manifest], Section 2.2.3.



Media Manifest provides the means to reference content either by asset identifier (physical asset) or by location.

2.3.1 Referencing assets by identifier

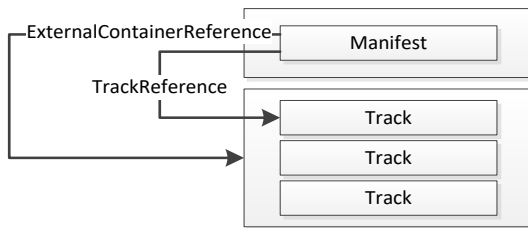
The most general and correct is to use `TrackIdentifier` in various Inventory types (Audio, Video, etc.). This allows assets to be referenced regardless of how the asset is containerized or packaged with other tracks. However, the Retailer's asset management system must be able to index off identifiers. Retailers might wish to consider this on their roadmaps.

2.3.2 Referencing assets by location

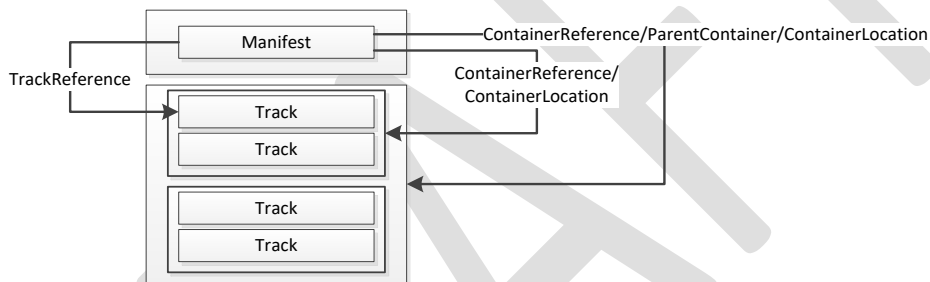
In this recommendation, the only location we are considering is filename. It is presumed that any other types of location (e.g., URL for download) will be resolved through the File Manifest that enumerates delivery mechanisms.

The element for referencing filenames is `ContainerReference` which is found under each of the Inventory media types (Audio, video, etc.). The outermost container in Inventory's `ContainerReference/ContainerLocation/...` should match the File Manifest's `FileInfo/Location`.

Consider the following diagram from [Manifest]. File Manifest FileInfo/Location corresponds with the Media Manifest's ExternalContainerReference.



The following illustrates track within nested containers. In this scenario, File Manifest's FileInfo/Location corresponds with the *outermost* container, in this case ContainerReference/ParentContainer/ContainerLocation. Note that the File Manifest does not address the internal structure of a container—the only possible reference is the file itself which is the outermost container.



2.3.3 Images

There are several types of images in a Media Manifest; for example, chapter images, metadata images and gallery images. It is important the recipient of a Media Manifest be able to easily locate images associated with an Experience.

All Images must be included in the Inventory.

Images that are part of an Experience include the following:

- Images in Picture Group referenced in Gallery/PictureGroupID
- Images referenced in BasicMetadata/LocalizedInfo/ArtReference. Basic Metadata exists in Experience and Audiovisual, both through reference (ContentID) and by inclusion (BasicMetadata).
- Images referenced in Presentation/Chapters/Chapter/ImageID for all Presentations referenced directly by the Experience, or referenced indirectly through a Playable Sequence

To make it easier to locate images associated with an Experience, the recommended practice is to also include images that belong together (e.g., chapter images and metadata images) in a Picture

Group. With the exception of Picture Groups referenced in Galleries, Picture Groups associated with an Experience should be referenced in Experience/PictureGroupID. Specifically

- All chapter images for a Presentation should be in a Picture Group
- All metadata images associated with a ContentID should be in Picture Group

References to images in BasicMetadata/LocalizedInfo/ArtReference should be to the Inventory. That is, the reference should be of form “md:imageid:“<scheme>“:“<SSID>. For example:

```
<manifest:Audiovisual ContentID="urn:dece:cid:eidr-s:C5CA-AD07-3B62-A2B0-23C1-G">
  <manifest:PresentationID>presentation</manifest:PresentationID>
  <manifest:BasicMetadata ContentID="urn:dece:cid:eidr-s:C5CA-AD07-3B62-A2B0-23C1-G">
    <md:LocalizedInfo language="en" default="true">
      ...
      <md:ArtReference resolution="800x600">md:imageid:eidr-x:C5CA-AD07-3B62-A2B0-23C1-
G:poster.en</md:ArtReference>
      ...
    </md:LocalizedInfo>
    <md:LocalizedInfo language="de">
      ...
      <md:ArtReference resolution="800x600"> md:imageid:eidr-x:C5CA-AD07-3B62-A2B0-23C1-
G:poster.de</md:ArtReference>
      ...
    </md:LocalizedInfo>
    ...
  </manifest:BasicMetadata>
</manifest:Audiovisual>
```

2.4 Determining Which Tracks Are Included in an Entitlement

A player only plays what a user has acquired. This section describes the process for mapping the acquisition to specific tracks. The goal is to ensure that a user gets the Experience associated with the entitlement (e.g., movie-only vs. movie with bonus features) and plays the tracks associated with the entitlement (e.g., ‘SD’ vs ‘HD’, and director’s commentary or not)

A player must determine whether individual tracks fall into the scope of an Avail. This is done by matching the disposition of an entitlement (e.g., the user purchased the title in ‘HD’) and information in the Avail for that entitlement with information in the Manifest.

This scenario assumes the user has acquired content in accordance with a particular Avail.

Selection is based on the following information from the Avail against which the content was acquired. This information is as follows:

- ALID – The ALID associated with the entitlement. ALID is in the Avail/ALID element.
- Condition – The current status of the entitlement. Condition is in the Transaction/ExperienceCondition element. Standard condition values are documented in [Manifest], Section 9.2. However, other conditions could exist.

- Media Profile (Format Profile) – Characteristics of the media in general terms (e.g., ‘HD’ and ‘SD’). Profile terms are document in [Avails], Section 2.2.3

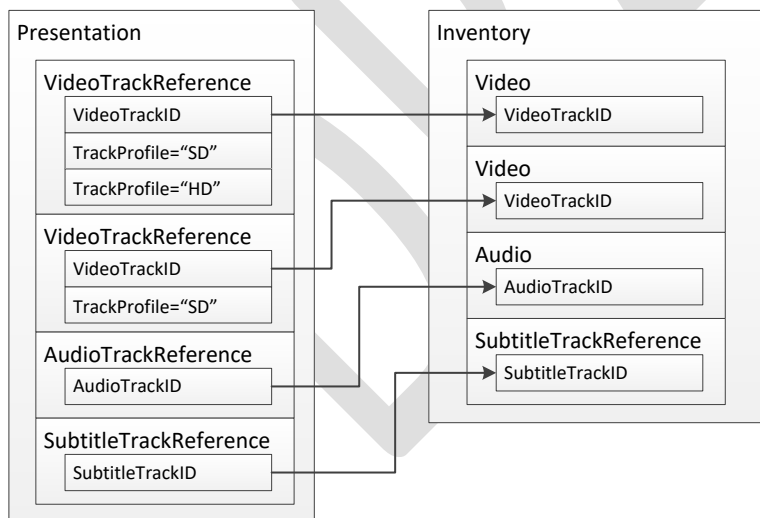
Given that information, and a properly constructed Media Manifest, the Player can determine which Experience to use and which tracks are covered by the entitlement.

First the player selects the correct Experience. ALIDExperienceMap allows ALID and Condition to map to a set of Experiences. That is an Experience maps if Avail/ALID from the Avail matches ALIDExperienceMap/ALID in the Manifest and Avail/Transaction/ExperienceCondition matches ALIDExperienceMap/ExperienceID/@condition for the Experience’s ID.

This set of Experiences is then downselected based on player settings such as Region and Language.

The Experience references one or more Presentation, possibly indirectly through Playable Sequence. Within that Presentation, a track is playable if its Profile matches the Avail. That is, the Avail’s Transaction/FormatProfile matches any instance of the track’s TrackProfile (TrackMetadata/AudioTrackReference/TrackProfile, .../VideoTrackReference/TrackProfile, etc.). Note that a Track that does not have any TrackProfile instance is assumed to match all profiles.

The following illustration includes a Presentation that includes both an HD and SD tracks, the HD track will have TrackProfile=‘HD’ and TrackProfile=‘SD’; and the SD track will have TrackProfile=‘SD’. If FormatProfile=‘HD’, both tracks are matched. If FormatProfile=‘SD’, only the SD track is matched. There is also an audio track and a subtitle track with no FormatProfile. The audio and subtitle tracks match all profiles.



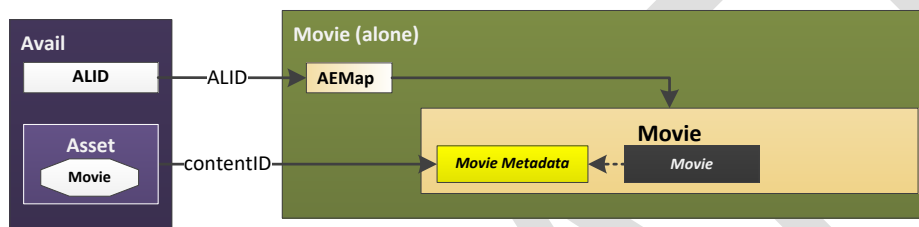
3 MOVIE

3.1 Movie

This is the most common case, but only for catalog. If a movie has not yet been delivered internationally, Regional Edits should be used.

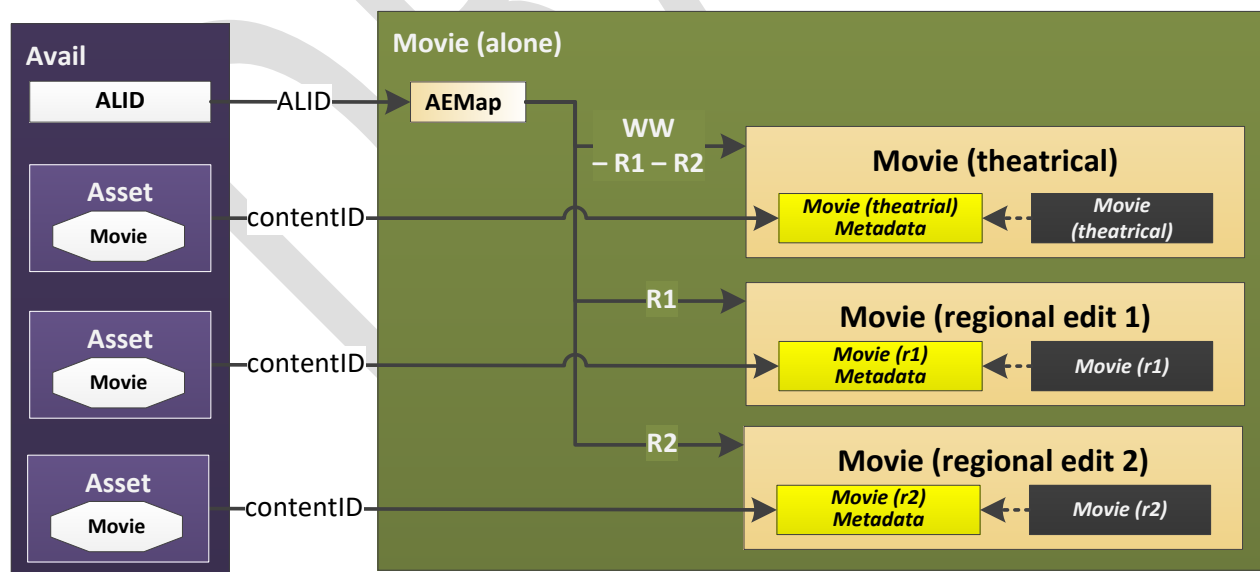
A single entry is in ALIDExperienceMap mapping the ALID to the ExperienceID for the movie.

The structure is the same for a single Episode. The following example is for a single episode, in this case Season 1, Episode 3. In general, each episode would get its own Avail and Experience.



3.2 Regional Edits

3.2.1 Static View (all edits known)

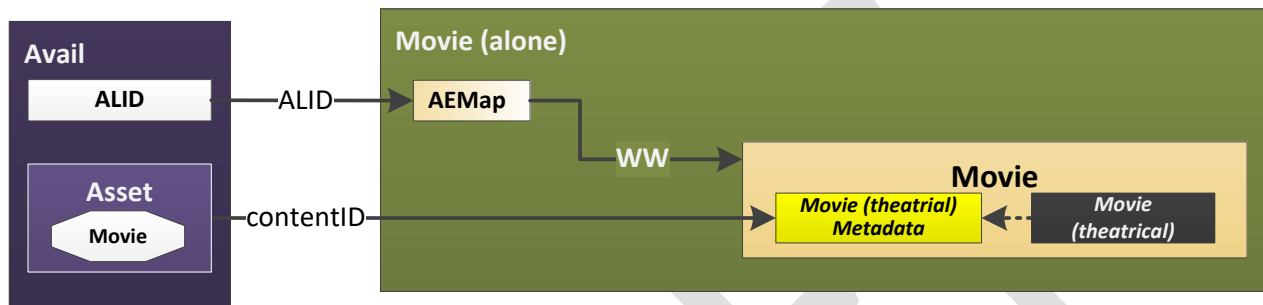


3.2.2 Dynamic View (edits arrive over time)

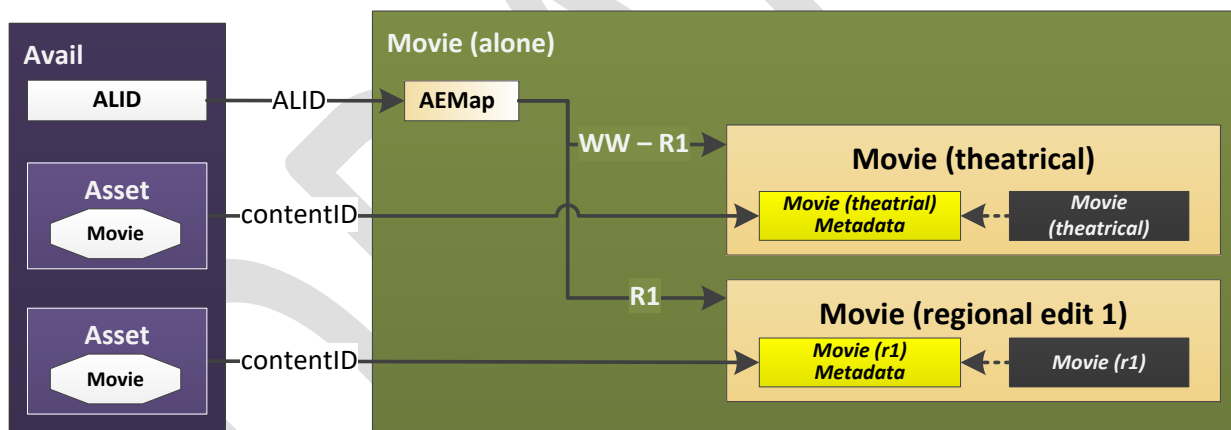
This covers the real-life scenario where Avails are offered before it's known whether or not a region will have a distinct edit. The ALID remains constant while Media Manifest (or other fulfillment data) responds to the new edits.

Given that the ALID is referencing one or more edits, best practice is to base the ALID on the EIDR Abstraction (Title-level).

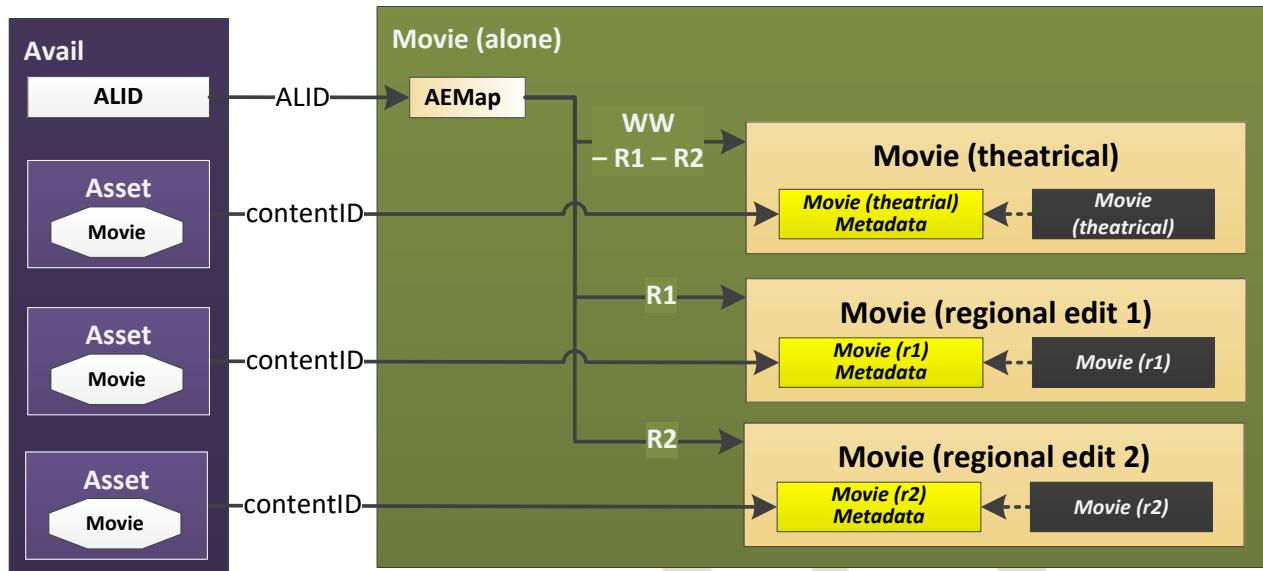
Initially, there is only one edit:



When an edit appears, the Manifest is updated to reflect the new edit.



This process is repeated for subsequent edits:

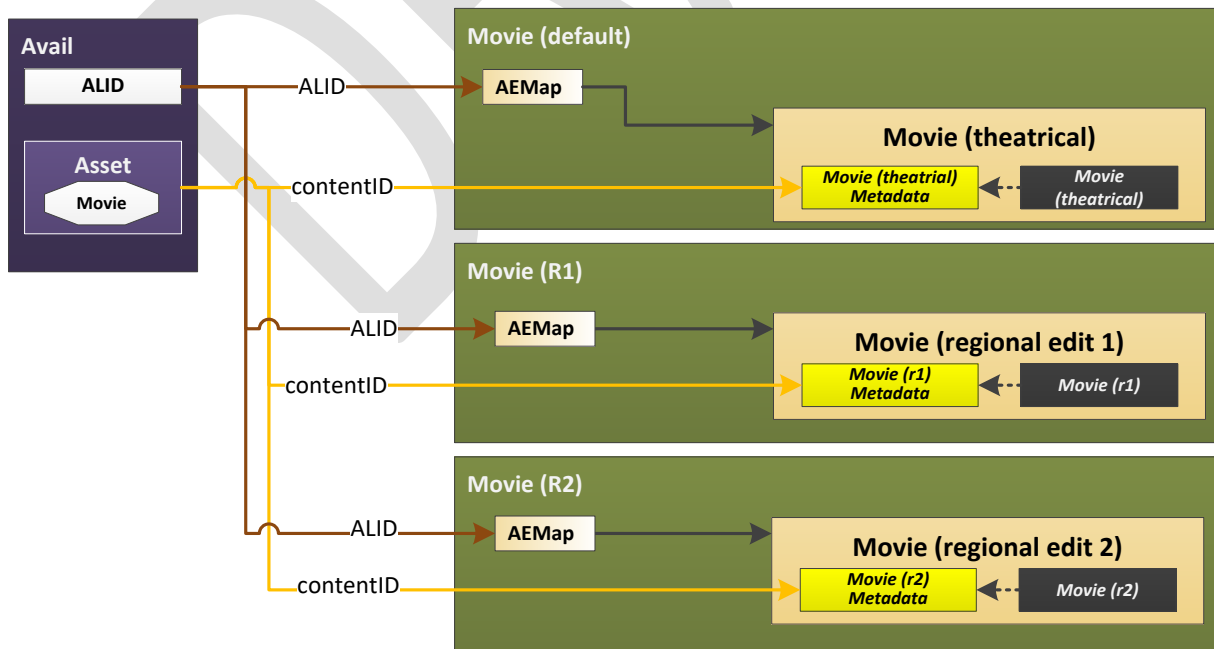


3.2.3 Regional Manifests

There is nothing requiring the same Manifest to be used in all regions. It's perfectly reasonable to send regional Manifests to specific regions.

However, this requires both the ALID and ContentID to be constant.

In the following example, everyone sees the same ALID and ContentID. Region 1 get the R1 Manifest. Region 2 gets the R2 Manifest, and everyone else gets the default Manifest. The existence of other Manifests should not be an issue in the given region.

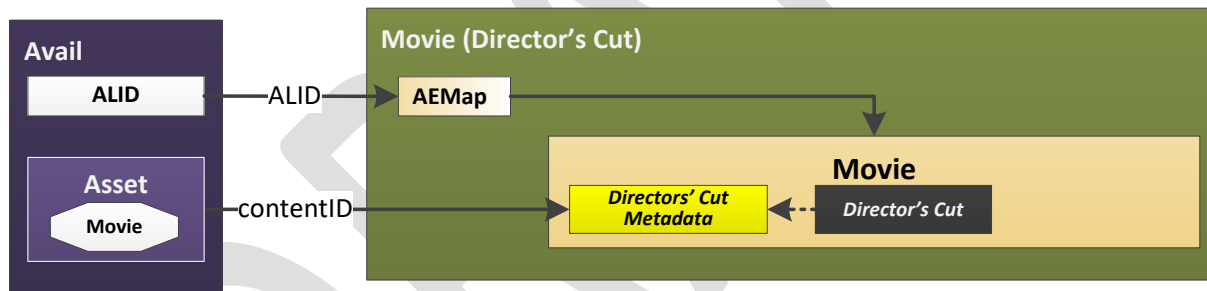
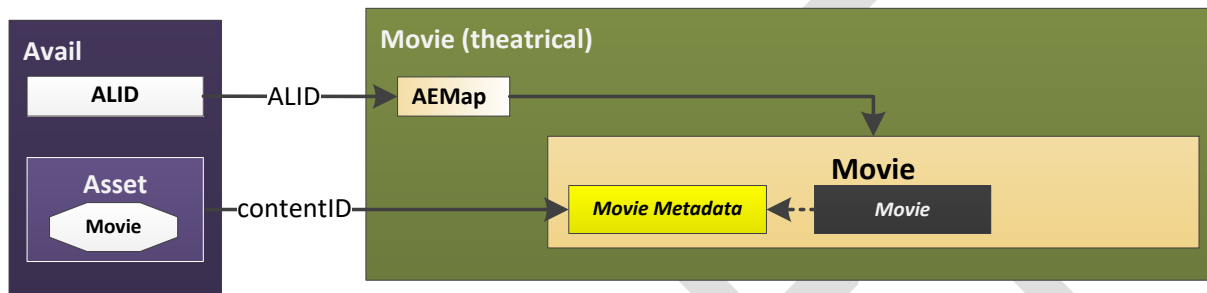


3.3 Director's Cut (or other special cut)

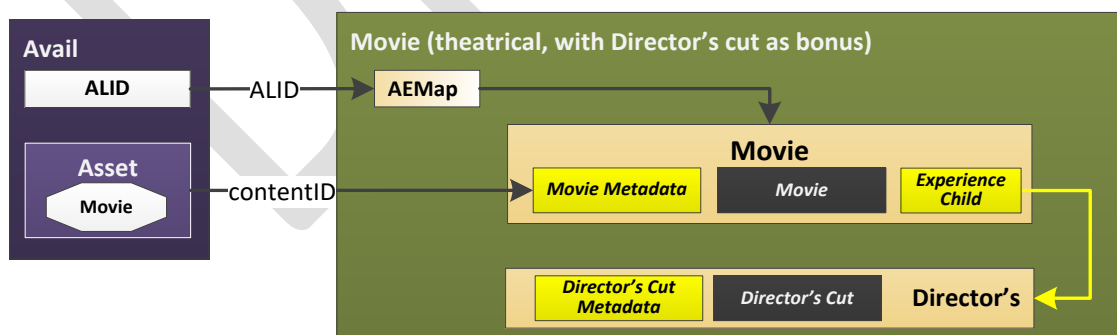
3.3.1 Distinct products

When a director's cut is offered as a distinct product, its encoding is identical to it being its own movie.

Assuming there will be no regional (ratings) edits on the director's cut, the ALID can be based on an EIDR edit. Otherwise, it should be managed in accordance with the Regional Edits practice.



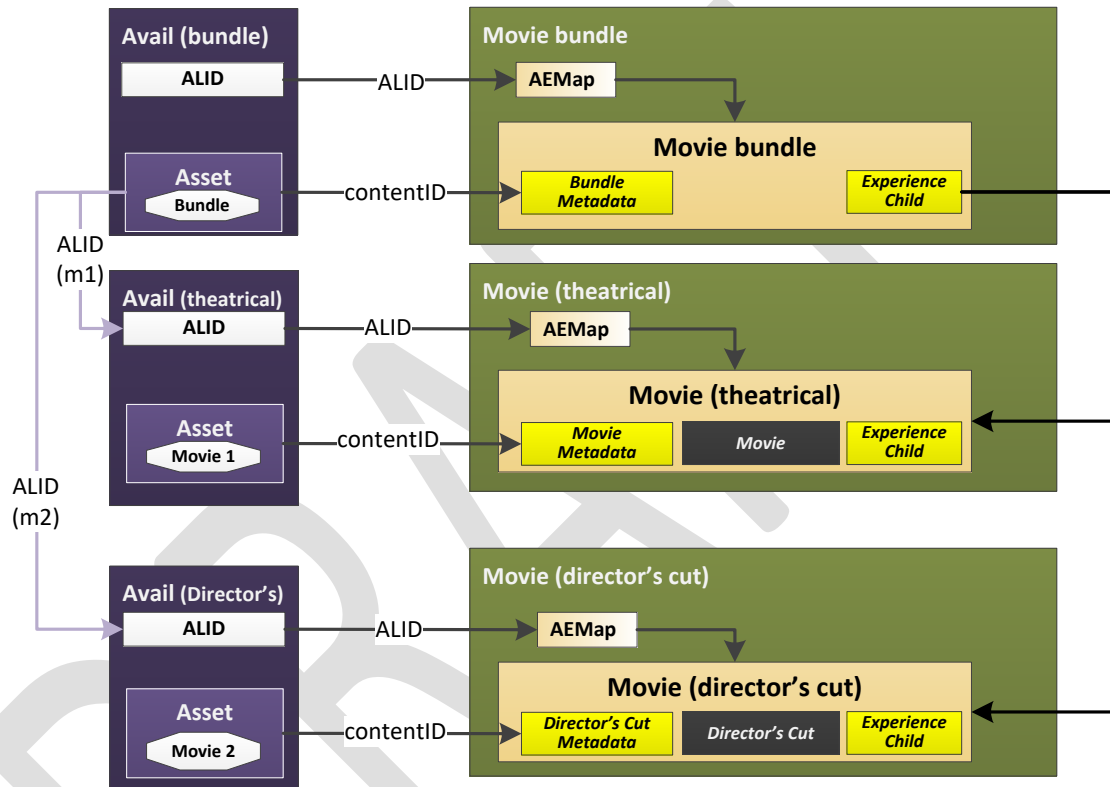
3.3.2 Directors' Cut as bonus



3.3.3 Director's Cut as Bundle

This scenario allows products to be sold separately or together. What's different about offering Director's cut as part of a Bundle is that distinct rights are maintained in the user's library. With one purchase, it's equivalent to buying the theatrical cut and the director's cut separately.

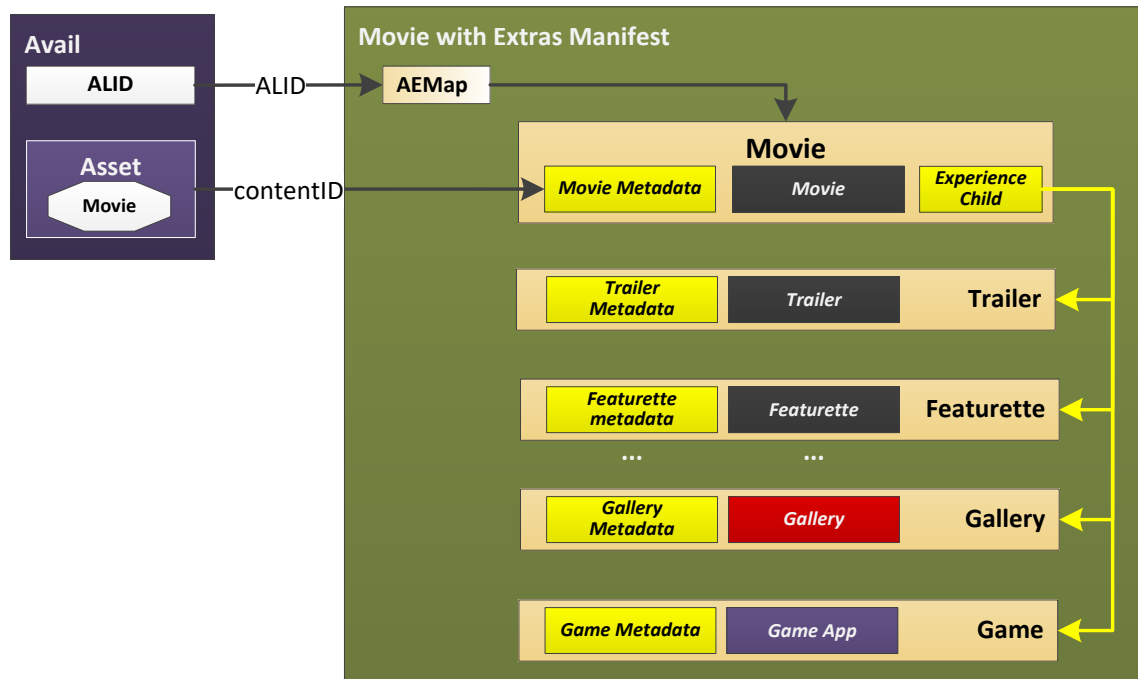
The most common uses of this structure are 1) When there is only one offering, and the user gets both, and 2) When there are two offerings (theatrical cut and Director's cut), but one offer gets the other (e.g., buy theatrical, get theatrical; buy Director's get Director's *and* theatrical).



3.4 Movie with Bonus

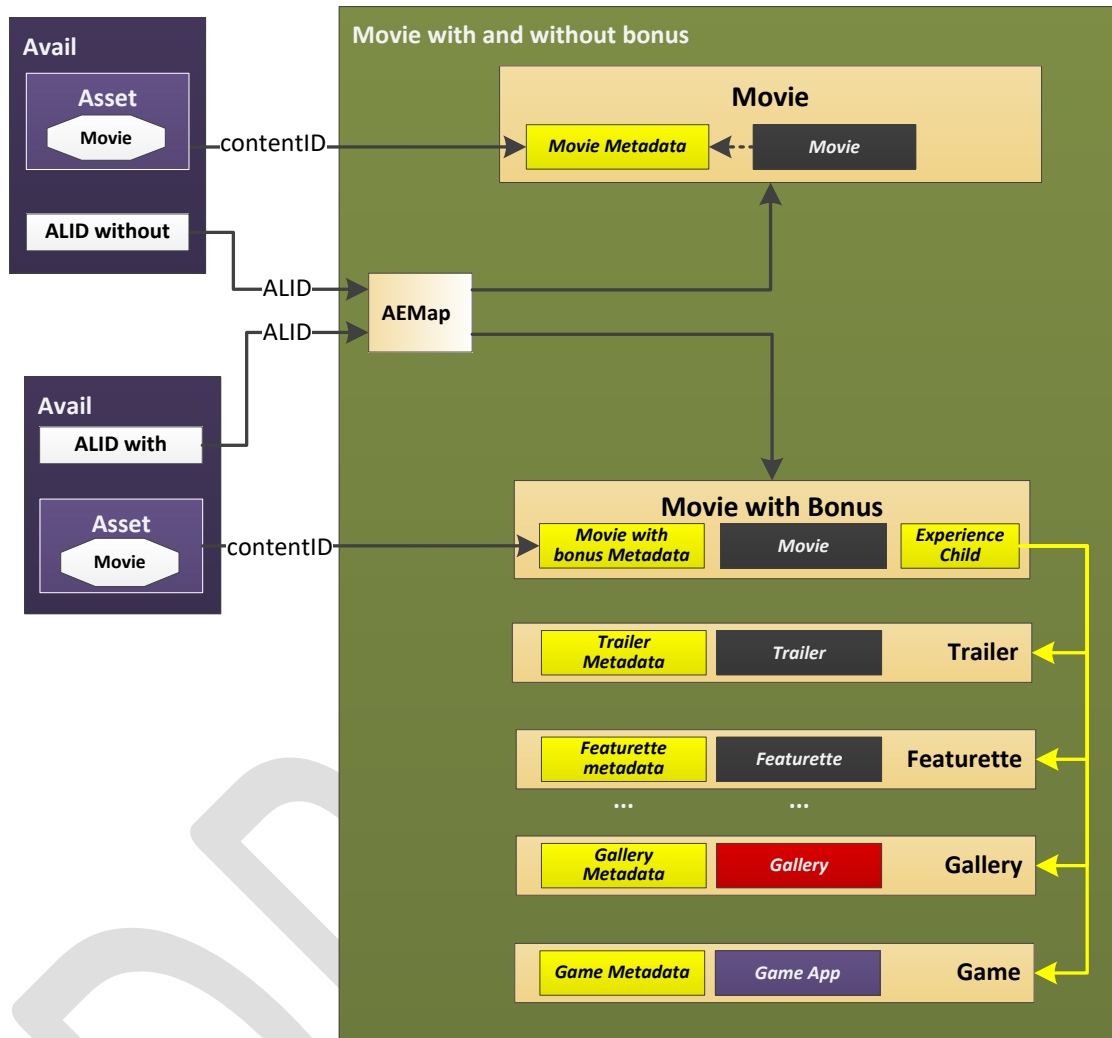
The following illustrates a movie with bonus material (extras).

Best practice is to derive the ALID from the movie's EIDR in EIDR-x form.



DRAFT

3.5 With and without bonus

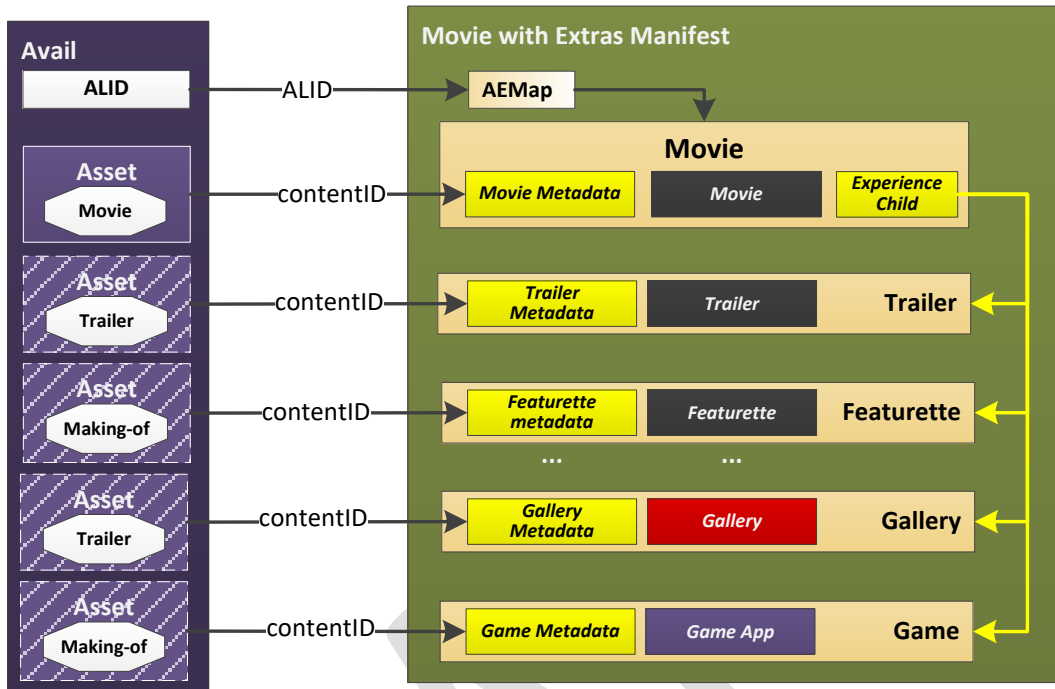


3.6 Movie with Extras, Avails reference Extras

The following movie example allows specific assets to be listed. This avoids any ambiguity about which assets are included. This presupposes that the asset list is constant or needs to be updated in the Avail. If the Media Manifest were updated without updating the Avail, there could be a disconnect.

As the assets are known, and EIDR compilation can be created for the ALID.

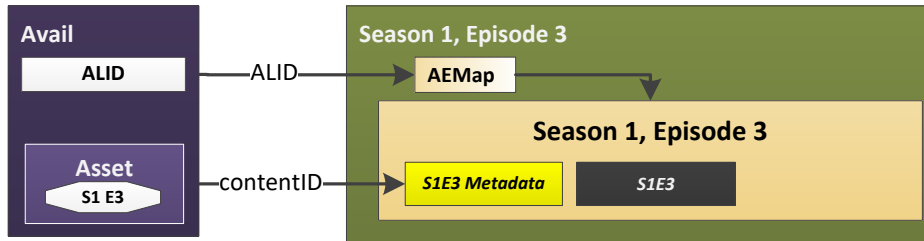
Asset elements are created for each Asset. In each of these Assets, metadata describes that individual Asset and the ContentID refers to metadata for that Asset. Note that these identifiers are for the individual Assets. Typically, these will be the EIDR Edit for that work.



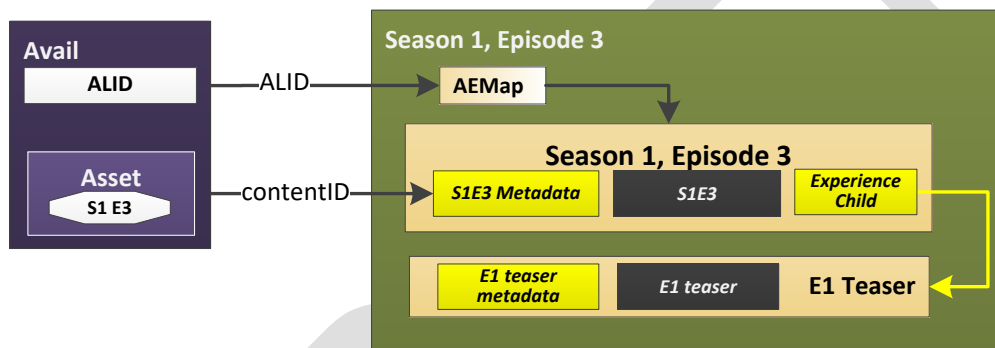
DRAFT

4 EPISODIC

4.1 Episode



4.2 Episode with Bonus

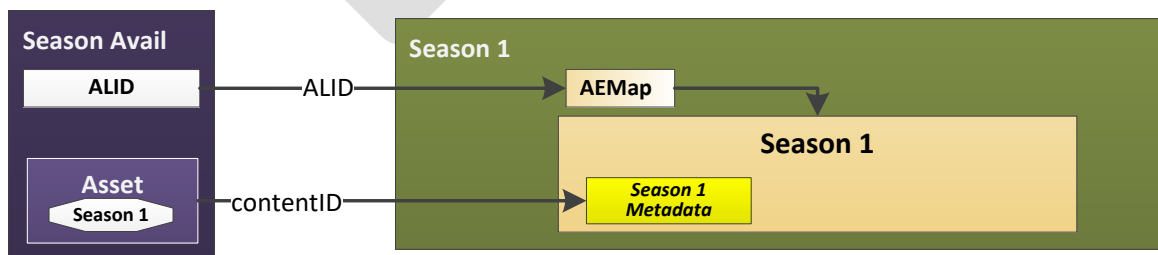


4.3 Season

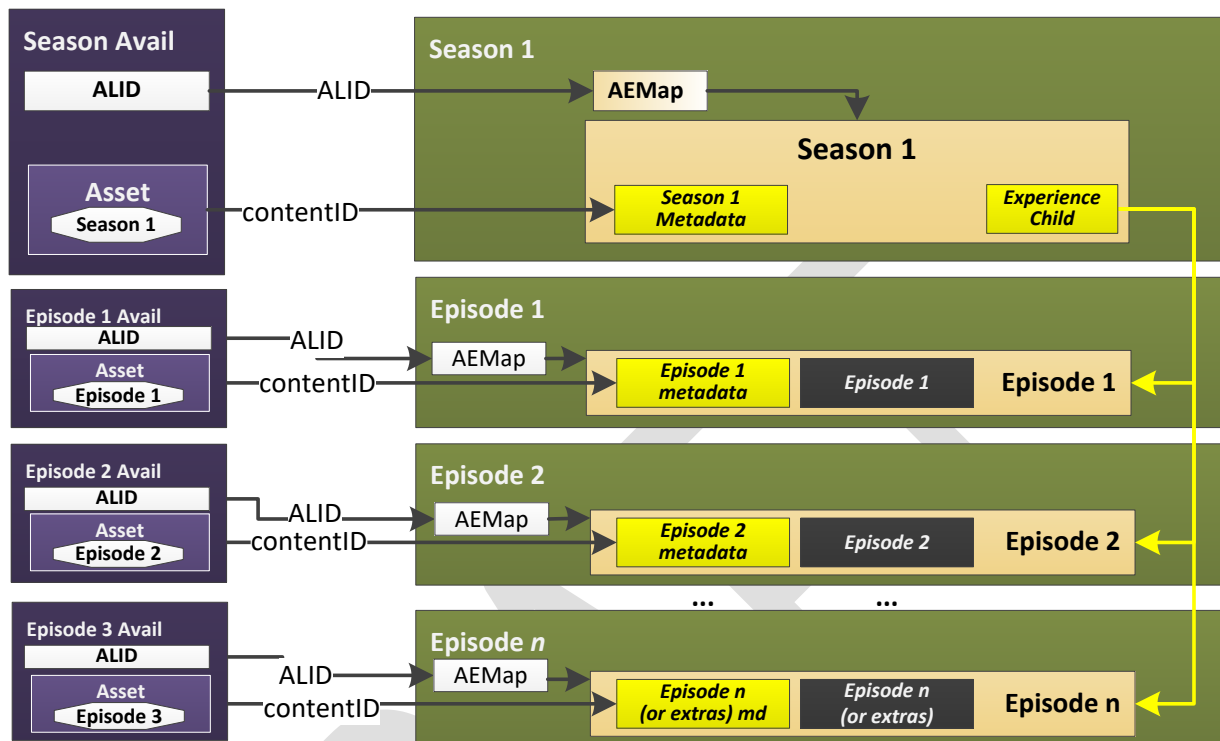
4.3.1 Without episodes listed (season pass, some or all episodes not aired)

The season without episodes explicitly listed as distinct Assets is similar to the Season without episodes listed as Assets (later example). However, since there is only one Asset (the season itself) and episodes might not yet be known, it does not make sense to reference individual episodes. It is understood from context that the Experience will ultimately include episodes. This model is used for presale (“Season Pass”) or a complete season.

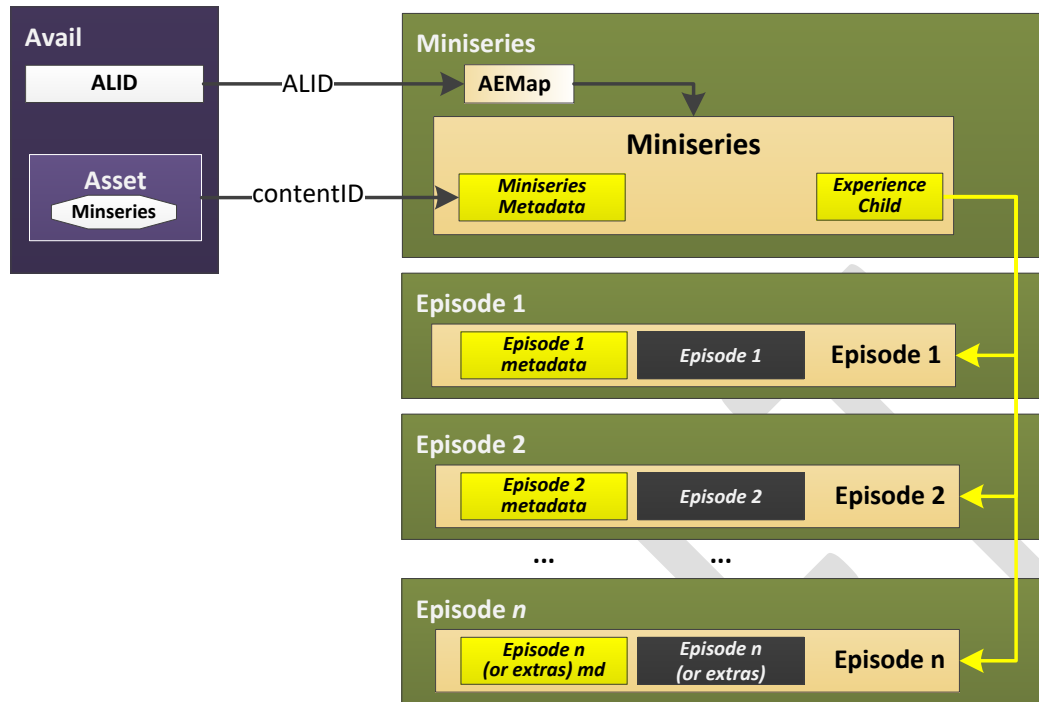
Note that the season metadata references the series metadata.



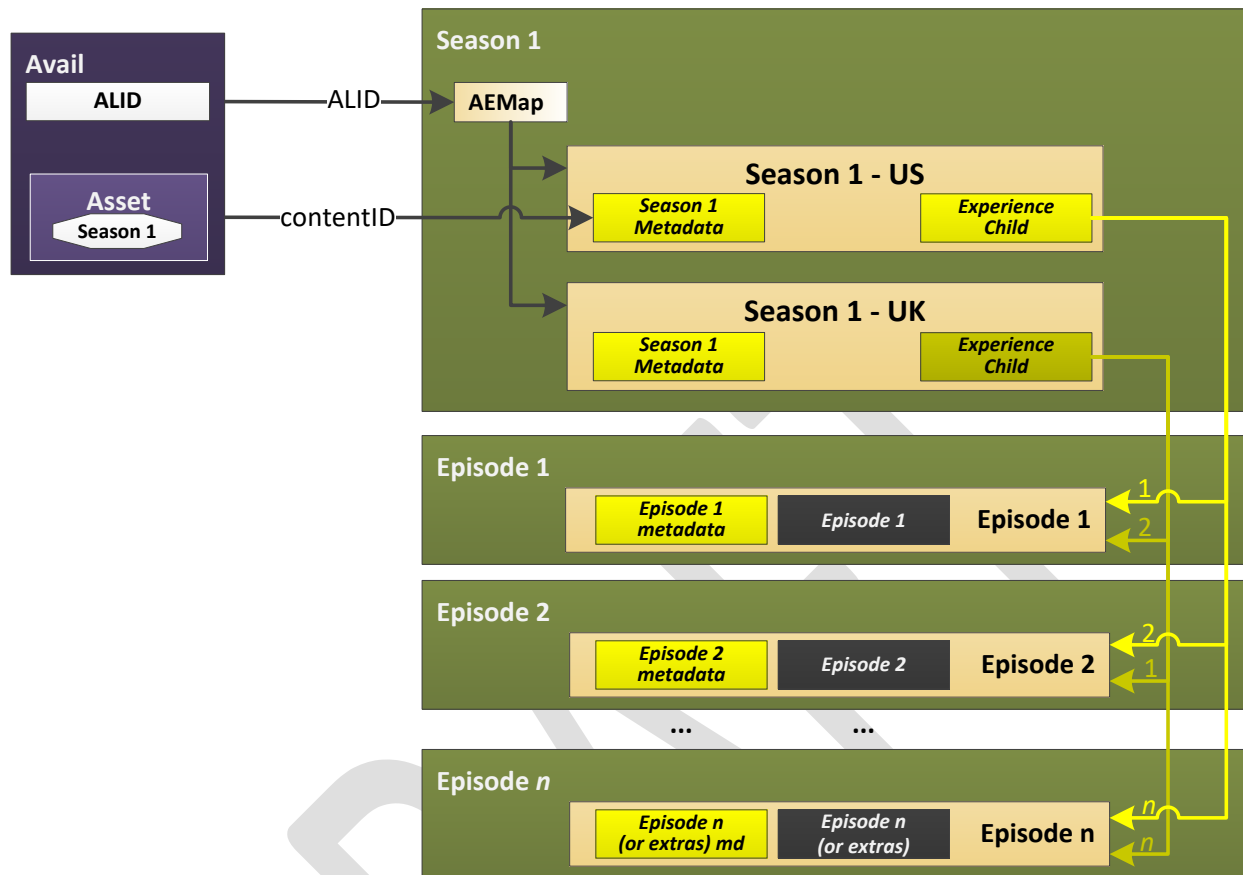
4.3.2 With episodes listed (season aired)



4.4 Miniseries



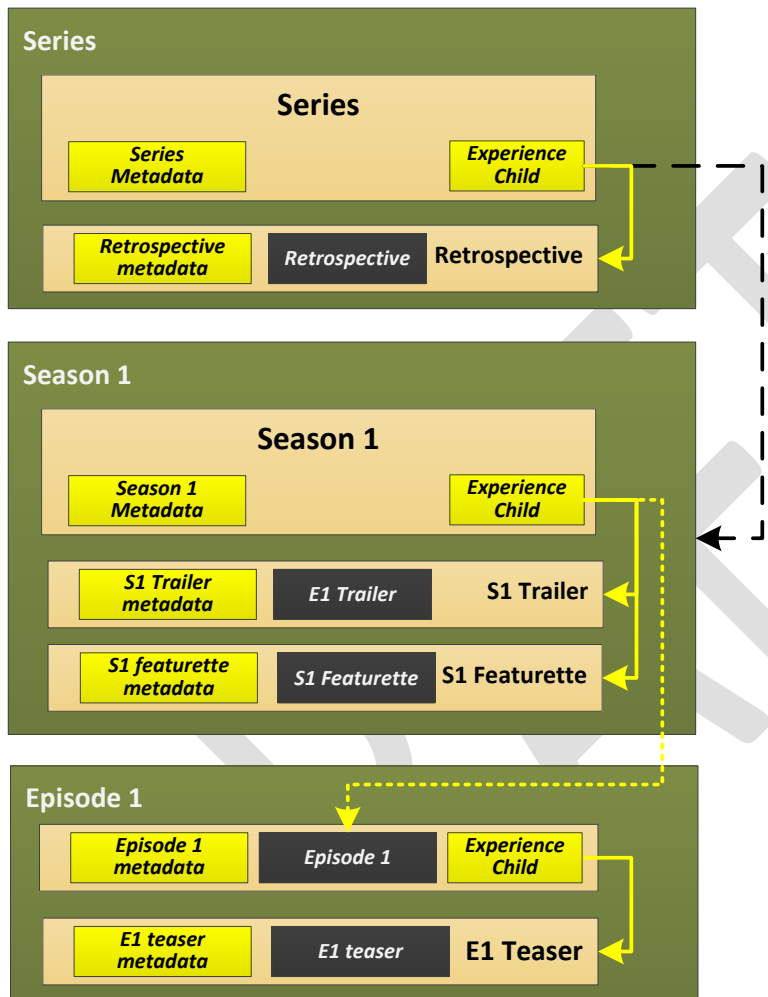
4.5 Alternate Episode Ordering



4.6 Series, Season and Episode with bonus

Avail is the same as Season.

[CHS: It's not clear how to find Series Experience.]



5 BUNDLES

These examples illustrate how an Avail can combine separate offerings into a Bundle.

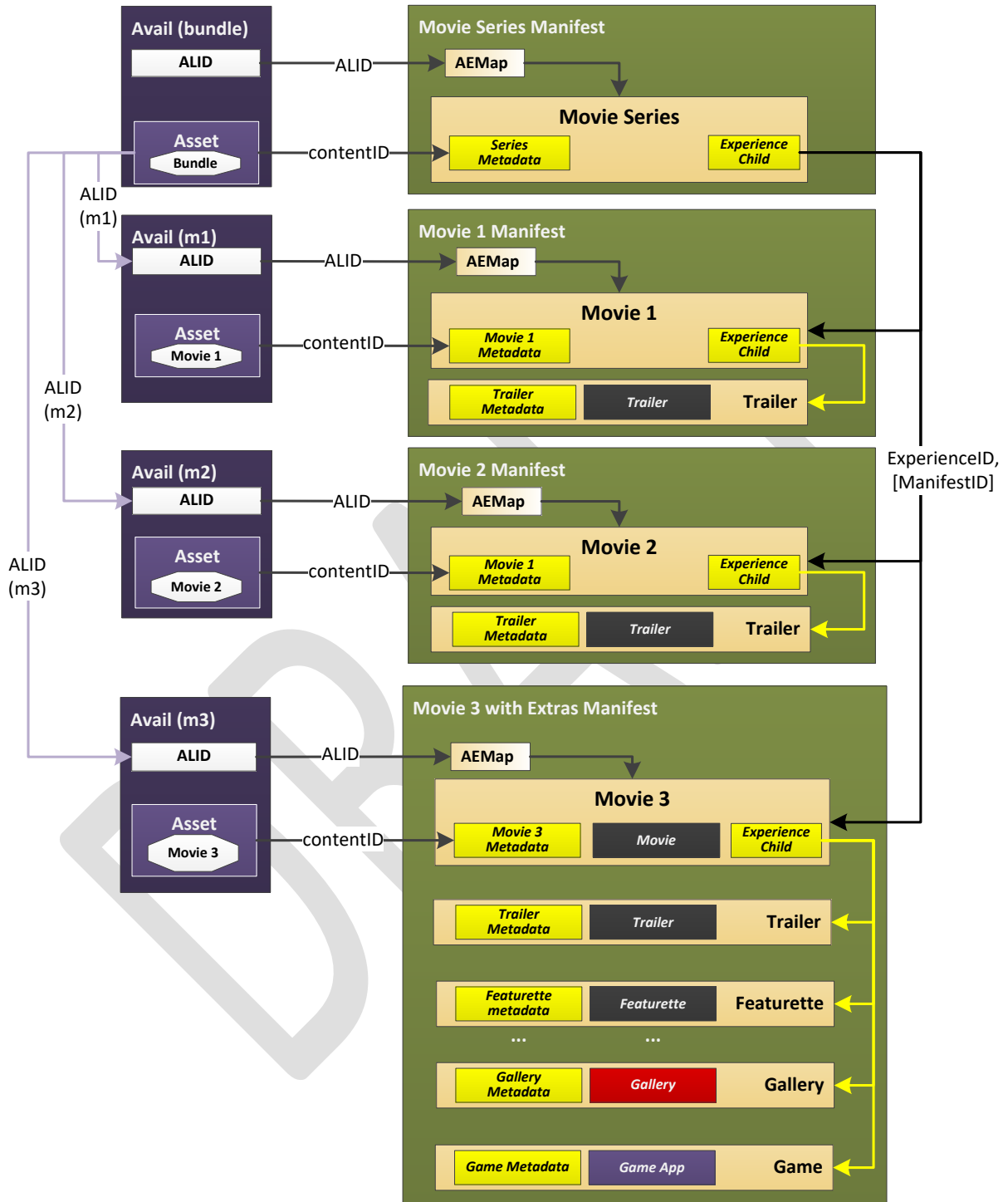
The Bundle is a distinct product, which means it has its own IDs, metadata and business terms. The Bundle, by definition, references content that would generally be sold separately—the digital equivalent of several Blu-rays shrink-wrapped together. This is achieved by referencing those other products.

The Bundle might even have some of its own content, such as exclusive bonus material available as part of the bundle.

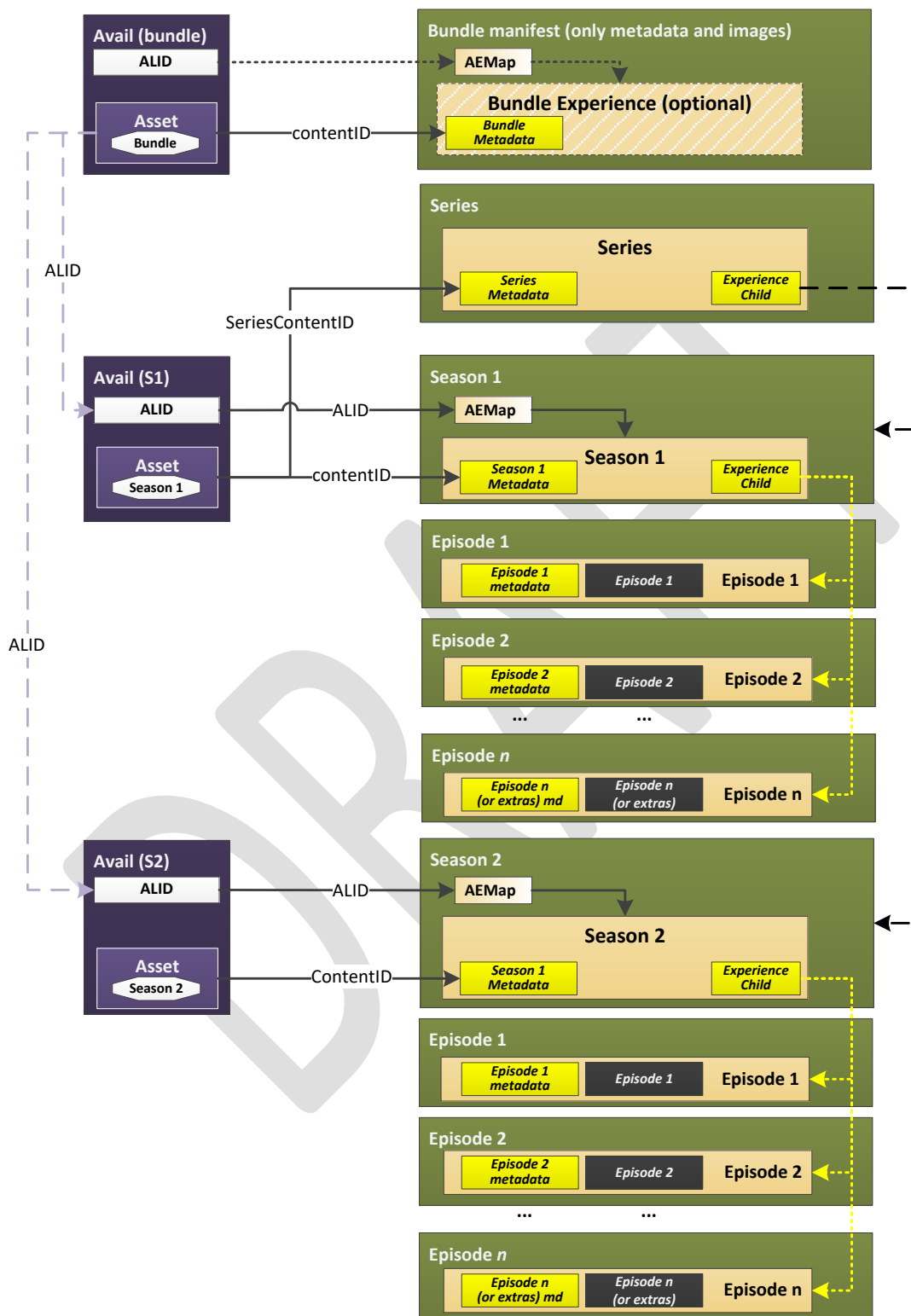
The following examples show movie bundles, TV bundles and mixed bundles. Bundles are not particular about what gets bundled. Hypothetically, anything referenceable by and ALID can be bundled, including other bundles (bundle of bundles)—think taking several shrink-wrapped bundles and putting them in a box.

DRAFT

5.1 Movie Bundle



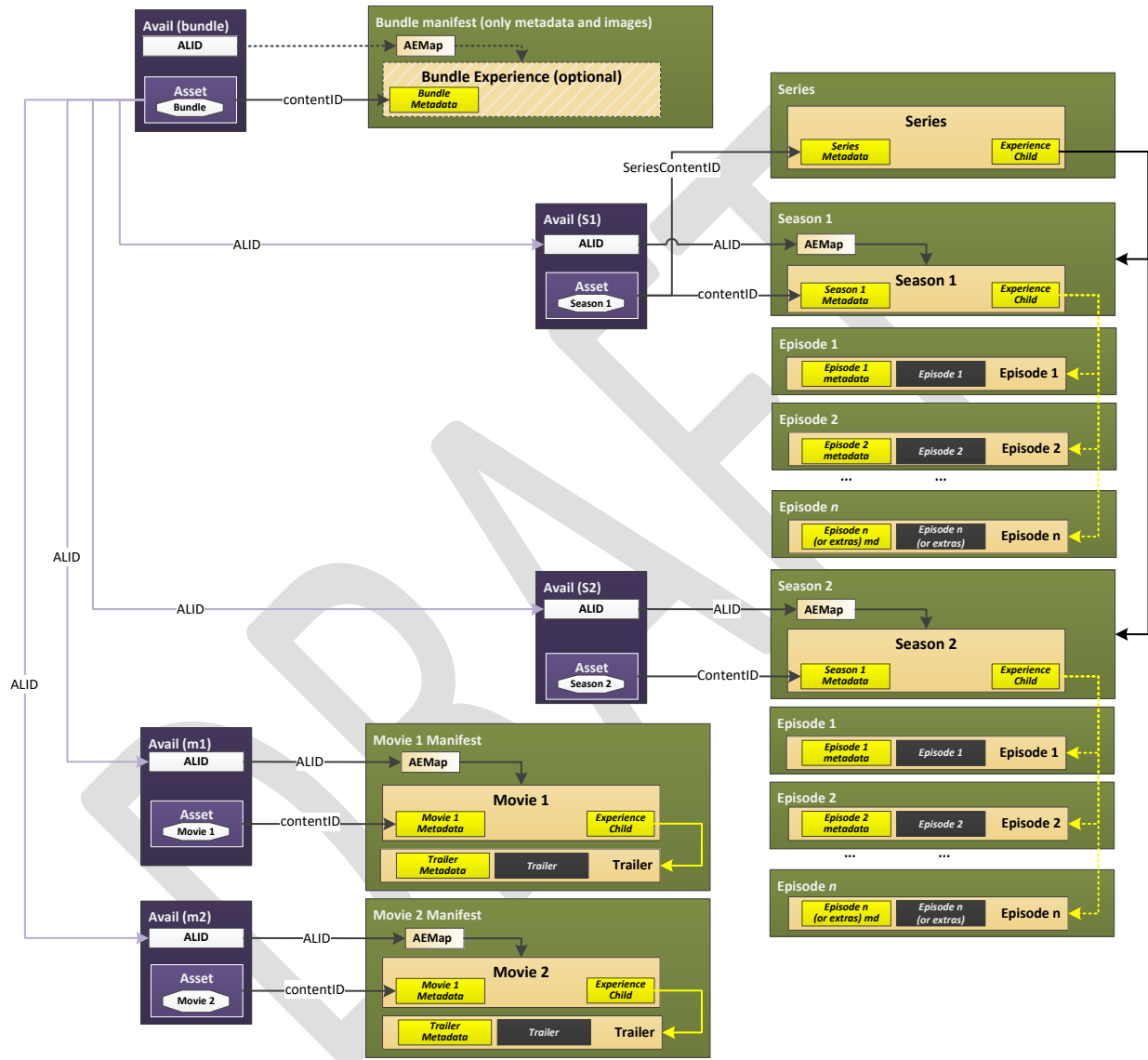
5.2 Season Bundle



5.3 Mixed Bundle

This Mixed Bundle bundles two TV seasons and two movies. The structure of the Bundle is no different from any other Bundle; only the content types.

As a Bundle of known content, an EIDR Compilation can be used as the ALID.



5.4 Bundle of Bundles

The Bundle of Bundles is similarly structured to the Mixed Bundle in that it just references other Bundles.

If the referenced bundles are EIDR compilations, an EIDR compilation can be created for a Bundle of Bundles.

